

ОПТИМИЗАЦИЯ КОММУНИКАЦИИ ВНУТРИ ВЕБ-ПРИЛОЖЕНИЯ С ПОМОЩЬЮ SIGNALR

А.Д. Соловьёва

Научный руководитель: В.В. Соколова, к.т.н.

Томский политехнический университет

E-mail: ads25@tpu.ru

Введение

В последнее время список сервисов и ресурсов, которые выросли в интернете небывалое множество. Интернет превратился из однообразных статических страничек в мощный инструмент интерактивности и общения с конечными пользователями. Во многих веб-проектах присутствуют элементы, значение и состояние которых необходимо часто менять. Возникает проблема: показывать ли актуальные значения после обновления страницы или же можно реализовать автообновление всех этих данных? В данном случае потребуются такое решение, которое позволит динамически менять элементы без лишней перезагрузки страницы и с минимальными ресурсозатратами. В данной статье будет рассмотрено, для чего же нужен SignalR и как он способствует решению данной проблемы на примере игрового веб-приложения под названием Cybertank.

Решение с помощью AJAX

Логика решения проблемы, изложенной выше, при помощи AJAX следующая: клиент с заданной периодичностью, опрашивает сервер, в поисках обновлений на странице, если сервер сообщил, что есть обновленные данные, то JavaScript внесён обновления в элементы страницы или отобразит уведомление. Но для больших проектов решение с AJAX, где каждый клиент будет опрашивать сервер, это создаст слишком большую нагрузку.

Решение с помощью SignalR

Если же используется SignalR, то клиент подписывается на серверный метод, который в случае изменения внутреннего состояния оповестит клиента об этом. Таким образом клиенту не требуется постоянно отсылать запросы на сервер, чтобы узнать, изменилось ли внутреннее состояние серверной части веб-приложения, сервер сам оповещает клиента о новых данных [1].

SignalR – это мощная высокоуровневая библиотека, которая абстрагирует множество базовых сложных технологий, чтобы обеспечить простой способ передачи данных между клиентом и сервером. SignalR управляет соединениями автоматически и позволяет отправлять данные, используя широковещательные или одноадресные передачи [2].

С помощью SignalR сервер может вызывать методы JavaScript на всех клиентах самостоятельно, когда требуются обновления.

Клиент JavaScript также может вызывать методы сервера. В случае, если можно использовать WebSocket, то он будет работать через WebSocket, если такой возможности нет, то он будет спускаться дальше, пока не найдёт приемлемый транспорт [3,4].

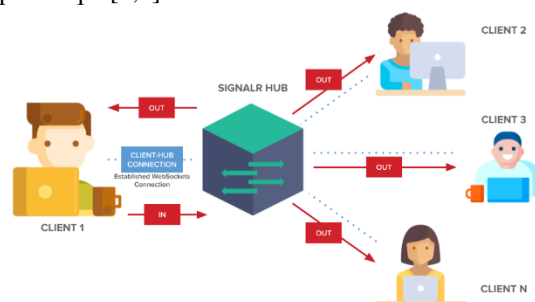


Рис. 1. Схема соединения SignalR

Где используется SignalR?

SignalR можно использовать везде, где пользователю потребуется обновить страницу, чтобы увидеть актуальные данные. Это позволяет серверу логически «передать» данные клиенту. Обычно это требуется для веб-панелей мониторинга и инструментов мониторинга, где информация должна постоянно обновляться без необходимости обновления страницы пользователем.

С этими возможностями SignalR идеально подходит для разработки приложений, таких как:

- Приложения чата
- Совместные приложения, такие как многопользовательские доски или инструменты управления проектами
- Онлайн игры
- Автоматическое обновление приложений, таких как информационные панели или карты в реальном времени
- Уведомления и оповещения [5]

Пример реализации

Перед IT компанией Sibedge встала задача по разработке командной игры, задачей которой является написать такого автоматического клиента (бота) для игры в Cybertank, чтобы он как можно чаще побеждал ботов других команд. Игра клиент-серверная, поэтому каждая команда получит логин для сервера, пользуясь которым бот сможет принимать участие в игре. Цели игры: создать интересный повод для привлечения сторонних разработчиков в компанию.

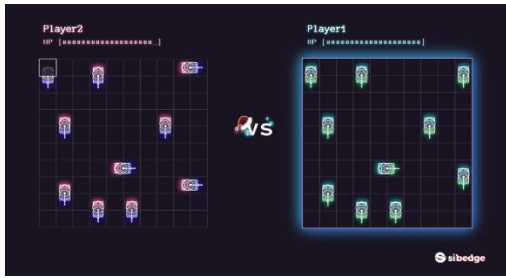


Рис. 2. Интерфейс игры

В качестве протокола взаимодействия между клиентской и серверной частью выбран API концентраторы SignalR ASP.NET, поскольку он позволяет выстроить событийно-ориентированный обмен сообщениями в режиме реального времени.

Коротко о том, как происходит взаимодействие: боты вызывают на сервере методы, куда отправляют расстановку танков. Далее подписываются на метод, который будет отображать все ответы от сервера, включая информацию о расстановке, и результатах выстрелов. Затем алгоритмы ботов в режиме очереди воспроизводят выстрелы. Как только на сервер пришла информация от ботов о расстановке или выстреле, он отправляет ее на Frontend-часть веб приложения. Frontend-часть подписывается на методы для приема расстановки и выстрелов и будет последовательно отображать данные на веб-форме. Обобщенное представление данного процесса отображено на рисунке 3 и в таблице 1.

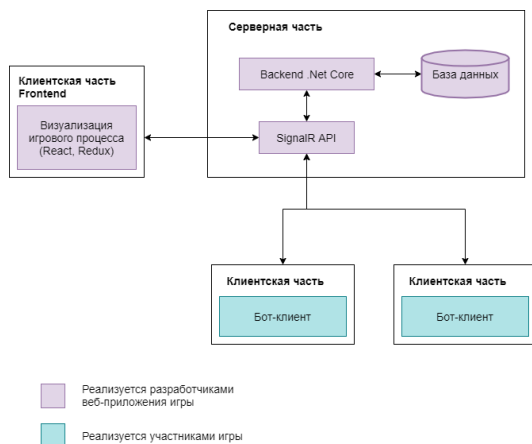


Рис. 3. Общая схема коммуникации

Таблица 1. Описание элементов схемы

№	Элементы схемы	Описание
Серверная часть		
1	База данных	База данных для хранения статистики по играм и участникам.
2	Backend.Net Core	Игровой сервер, реализующий логику работы игры.
3	SignalR API	Средства обмена данными между сервером и ботом-клиента, а также между

		серверной и клиентской частью приложения.
Клиентская часть		
4	Бот-клиент	Программа-клиент, разработанная участником для осуществления обращений к серверу игры в целях выиграть игру по заданному в ней алгоритму.
5	Frontend – Визуализация игрового процесса	Клиентская часть приложения для визуализации игр бот-клиентов.

Такая реализация позволяет каждому компоненту системы быть осведомленным о внутренних процессах того компонента, от которого она зависит и своевременно менять свое состояние в соответствии с этим.

Вывод

В данной статье были рассмотрены преимущества использования SignalR перед AJAX и выявлено, что делает эту библиотеку такой мощной и универсальной. SignalR отлично подходит для реализации надежных коммуникаций в реальном времени, которые хорошо работают в веб, настольных и мобильных приложениях.

Список использованных источников

1. Realtime Technology Glossary. [Электронный ресурс]. – URL: <https://www.pubnub.com/learn/glossary/what-is-signalr/> (дата обращения 09.01.2020).
2. Gustavo Armenta Valdez (December 12, 2012). «SignalR: Building real time web applications». Microsoft MSDN.
3. Real-time ASP.NET with SignalR. [Электронный ресурс]. – URL: <https://dotnet.microsoft.com/apps/aspnet/signalr/> (дата обращения 11.01.2020).
4. What is SignalR and why should I use it? [Электронный ресурс]. – URL: <https://blog.cindypotvin.com/what-is-signalr-and-why-should-i-use-it/> (дата обращения 15.01.2020).
5. An Introduction to Real-Time Communication with SignalR. [Электронный ресурс]. – URL: <https://www.red-gate.com/simple-talk/dotnet/aspnet/an-introduction-to-real-time-communication-with-signalr/> (дата обращения 11.01.2020).